UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,216	10/22/2003	Richard A. Han	10806.00	9545

26889 7590 05/04/2009

MICHAEL CHAN
NCR CORPORATION
1700 SOUTH PATTERSON BLVD
DAYTON, OH 45479-0001

EXAMINER	
MOORTHY, ARAVIND K	
ART UNIT	PAPER NUMBER
2431	

MAIL DATE	DELIVERY MODE
05/04/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/691,216
Filing Date: October 22, 2003
Appellant(s): HAN ET AL.

Peter H. Priest
Reg. No. 30,210
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed February 11, 2009 appealing from the Office action mailed July 15, 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

7,249,262	Hauck et al	07-2007

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-10 and 12-20 are rejected under 35 U.S.C. 102(e) as being anticipated by Hauck et al U.S. Patent No. 7,249,262 B2 (hereinafter Hauck).

As to claim 1, Hauck discloses a system for a licensee to control access to or distribution of software and/or data among a plurality of client nodes, the system comprising:

> means for storing software and/or data that is to be made available to pre-determined licensed client nodes (i.e. A small " client-side" software application, or client tray application, must be installed and run on each client machine that is to have access to protected Web pages maintained by server 16 before such client machine can access protected Web content) [column 5 line 65 to column 6 line 2], each client node of the plurality of client nodes (i.e. client #1, client #2 and client #3 in figure 1) being a data processing device for which access to specified software or data may be allowed if licensed (i.e. a given client machine can be authorized to access two or more different subscription services) [column 6, lines 32-35], and for storing a list of identifiers for licensed client nodes, each identifier

uniquely identifying one of the predetermined nodes (i.e. the temporary storage table application software is multi-thread capable and maintains a list of temporarily authorized clients) [column 10, lines 28-30], the presence of each identifier on the list authorizing the predetermined client node associated with the identifier to be allowed access to the software and/or data (i.e. Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32]; and

a client application at each client node, the client application performing authentication taking place at the client node, authentication being accomplished by comparing the client identifier for the node against the list and allowing or rejecting access to the software and/or data by the client node at which the client application resides based on evaluation by the client application at the client node as to whether the identifier of the client node appears in the list (As discussed above, Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that

generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32].

As to claim 2, Hauck discloses that the means for storing the software and/or data and the list of unique client identifiers is portable [column 6, lines 15-21].

As to claim 3, Hauck discloses that the means for storing the software and/or data and the list of unique client identifiers comprises a compact disc [column 6, lines 15-21].

As to claim 4, Hauck discloses that the means for storing the software and/or data and the list of unique client identifiers comprises a floppy disc [column 6, lines 15-21].

As to claim 5, Hauck discloses that the client nodes are part of a communications network [column 5, lines 16-20].

As to claim 6, Hauck discloses that the means for storing is provided in a shared information storage area of a server that can be remotely accessed by at least some or all of the client nodes [column 5, lines 16-20].

As to claims 7, 14 and 17, Hauck discloses that the client application is operable to generate a unique identifier for the client node on which the client application resides and compare this with the unique identifiers on the authorized list, thereby to identify whether the unique identifier for that node is on the list [column 7, lines 23-32].

As to claims 8, 12, 15 and 18, Hauck discloses that the client executes a license management program which uses node specific data to generate the unique identifier [column 7 line 53 to column 8 line 13].

As to claim 9, Hauck discloses a method for a license to control access to or distribution of software and/or data among a plurality of client nodes, the method comprising:

> storing in association with the software and/or data, a list of unique identifiers for licensed client nodes, each of which uniquely identifies one of the nodes authorized to be allowed access to the software and/or data (i.e. Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32];
>
> identifying at each node whether a unique identifier for a particular node is included on the list [column 7, lines 23-32]; and
>
> controlling the operation of each node so that the list is examined at each node and the unique identifier is compared against the list, and loading, installation, or use of the software and/or data is allowed or rejected based on the

comparison at the client node of the unique identifier against the list [column 7, lines 23-32].

As to claim 10, Hauck discloses a program storage device, readable by a machine, having encoded thereon instructions executable by the machine for:

executing a license management program to establish a unique identifier associated with the machine executing the instructions (i.e. Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32];

reading a list of unique identifiers associated with specified software and/or data, each unique identifier being uniquely associated with one of a plurality of machines and establishing its associated machine as licensed for the specified software and/or data [column 7, lines 23-32]; and

controlling the operation of a client node comprising the machine executing the instructions so as to allow or reject access by the machine to the software and/or data based on a comparison taking place at the client node of the

unique identifier for the client node against the list of unique identifiers [column 7, lines 23-32].

As to claim 13, Hauck discloses a data processing device serving as a client node comprising:

means for reading a list of unique identifiers associated with software and/or data, each unique identifier being uniquely associated with one of a plurality of client nodes or terminals licensed to use the software and/or data (i.e. Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32]; and

means for controlling the operation of the data processing device so that the data processing device examines its own unique identifier and the list of unique identifiers and allows or rejects loading, installation, or use of the software and/or data based on a comparison taking place at the data processing device of its own unique identifier against the list of unique identifiers [column 7, lines 23-32].

As to claim 16, Hauck discloses a self-service terminal comprising:

means for reading a list of unique identifiers associated with software and/or data, each unique identifier being uniquely associated with one of a plurality of self-service terminals licensed to use the associated software and/or data (i.e. Each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to herein as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted) [column 7, lines 23-32]; and

means for controlling the operation of the self-service terminal so that the self-service terminal examines a unique identifier associated with the self-service terminal and the list of unique identifiers and allows or rejects loading, installation, or use of the software and/or data based on a comparison taking place at the self-service terminal of the unique identifier of the self-service terminal against the list of unique identifiers [column 7, lines 23-32].

As to claim 19, Hauck discloses that the self-service terminal is an automated teller machine in a network comprising a plurality of automated teller machines operated by a common operator licensed to utilize the associated software and/or data [column 5, lines 20-24].

As to claim 20, Hauck discloses that the automated teller machine executes license management software to generate its associated identifier (i.e. The client machine key DLL (typically generated by the Web site administrator using SoftSentry software) analyzes hardware characteristics of a particular local computer, or client machine (including hard drive characteristics, RAM characteristics, input/output device parameters and other hardware specific details), and thereby generates an integer value, or machine-specific identifier that is virtually unique for a given client machine. The client machine key DLL then performs an algorithm to calculate a password, based upon the machine-specific identifier, valid only for a client machine having the same virtually unique system identifier. As a result, even if the client tray application (and/or client machine key DLL) is copied by a user to a second client machine, the client machine key DLL on the second client machine will report a new machine-specific identifier that is almost certainly different from the machine-specific identifier that was generated by the user's first client machine) [column 7 line 53 to column 8 line 13]. Hauck discloses that the list of unique identifiers licensed to utilize the associated software and/or data is generated at a remote server which communicates with the automated teller machine over the network [column 7 line 53 to column 8 line 13].

(10) Response to Argument

On page 6, the Appellant argues, regarding claim 1, authentication takes place at the client node, in which the appearance of the client node's identifier on a list of identifiers of authorized clients is examined by a client application at the client node in order to determine whether the client node will allow access to software or data. The Appellant argues that Hauck

does not teach and does not make obvious such authentication, but instead teaches examination by a server of a storage table for the appearance of a session identifier submitted by a client.

The examiner respectfully disagrees. Hauck discloses that each protected Web site (or each protected "content cluster" of each Web site) has a unique dynamic link library (DLL) associated therewith. This DLL is referred to as a "client machine key", and it is this DLL that generates the machine-specific identifier for the user's client machine. This client machine key DLL provides a virtually unique client machine identifier, and includes an algorithm for ensuring that the user enters a password uniquely corresponding to the unique client machine identifier before access to protected data is granted. If a user desires access to such protected Web site, or to the protected "content cluster", then the specific Web site DLL must exist on the user's client machine [column 7, lines 23-35]. The examiner asserts that the machine-specific identifier would be the client node identifier. The client tray application, together with the client machine key DLL, determines the identity of the client-machine and other unique information including the "authorization rights/status of the client-machine" relative to the current server or content cluster "being accessed [column 7, lines 53-57].

On page 6, the Appellant argues, regarding claim 1, Hauck does not teach that the client side application controls access to protected content, and does not teach that this client side application compares the client machine identifier against a list before allowing access to content. The Appellant argues that Hauck makes it clear that the server receives information from a client machine and compares the client machine information against a storage table in granting or denying requests for content.

The examiner respectfully disagrees. As discussed above, the client tray application, together with the client machine key DLL, determines the identity of the client-machine and other unique information including the "authorization rights/status of the client-machine" relative to the current server or content cluster "being accessed [column 7, lines 53-57]. The step of generating machine-specific identification is represented in FIG. 2A by block 38. In the preferred embodiment, this step is achieved by using a software application package known as SoftSentry software, or a similar application, in conjunction with the client tray application. The client machine key DLL (typically generated by the Web site administrator using SoftSentry software) analyzes hardware characteristics of a particular local computer, or client machine (including hard drive characteristics, RAM characteristics, input/output device parameters and other hardware specific details), and thereby generates an integer value, or machine-specific identifier that is virtually unique for a given client machine. The client machine key DLL then performs an algorithm to calculate a password, based upon the machine-specific identifier, valid only for a client machine having the same virtually unique system identifier. As a result, even if the client tray application (and/or client machine key DLL) is copied by a user to a second client machine, the client machine key DLL on the second client machine will report a new machine-specific identifier that is almost certainly different from the machine-specific identifier that was generated by the user's first client machine [column 7 line 57 to column 8 line 13].

On page 7, the Appellant argues that Hauck teaches a password appropriate to the client machine identifier is computed and transmitted appropriately. A user entering appropriate subscription information is given a corresponding password by a server and is prompted to enter the password by the client side application. Upon entry, the entered password is compared with

the transmitted password and a failure results in denial of access to content. The Appellant argues that this comparison does not involve comparison by a client application of a client machine identifier against a list, and a successful entry is what leads to the step of entering client machine information in a list against which comparisons are made.

The examiner respectfully disagrees. As discussed above, the examiner asserts that the machine-specific identifier would be the client node identifier. The client tray application, together with the client machine key DLL, determines the identity of the client-machine and other unique information including the "authorization rights/status of the client-machine" relative to the current server or content cluster "being accessed [column 7, lines 53-57].

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Aravind Moorthy

/Aravind K Moorthy/

Examiner, Art Unit 2431

Conferees:

Christopher Revak

/Christopher A. Revak/

Primary Examiner, Art Unit 2431

/Ayaz R. Sheikh/

Supervisory Patent Examiner, Art Unit 2431